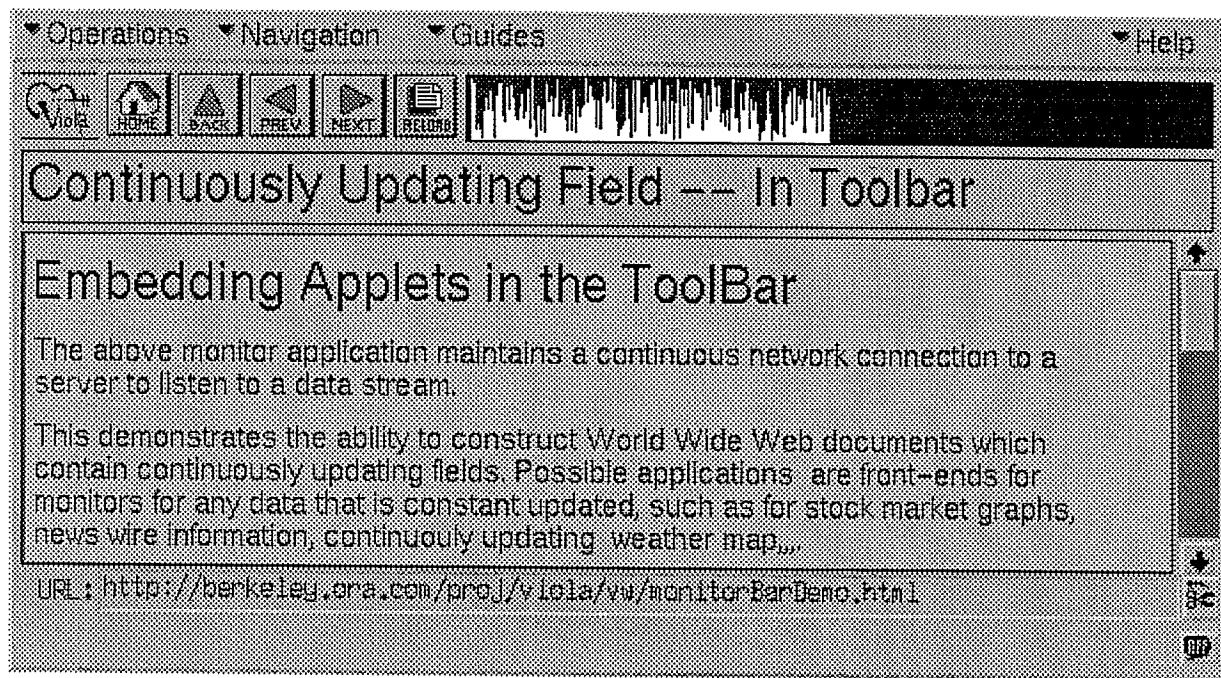


# Embedding Viola Applets in the ToolBar

## Example: Continuously Incoming Data Monitor



The above "monitor" applet (the scrolling field next to the "reload" button") maintains a continuous network connection to a server and listens to a data stream.

This demonstrates the ability to construct World Wide Web documents which contain continuously updating fields. Possible applications are front-ends for monitors for any constant updated data, such as for stock market graphs, news wire information, continuously updating weather map,,,.

## How does this work?

Basically, this document has a "tool" type <LINK> to a script. This causes viola to fetch the script, instantiates the script (objects), and installs the object in the tool bar.

(<LINK> is a HTML 3.0 provision for linking to data types)

This is the markup, embedded in a HTML document, that makes the link to the applet:

```
<LINK REL="tool" HREF="monitorBar.v">
```

And here's the listing for the "monitorBar.v" applet/object/script.

```
\name {monitorBar}
```

```

\children {monitorBar.socket}
\class {field}
\script {
    switch (arg[0]) {
        case "graph":
            /* draw the latest graph line, and scroll the field */
            y = arg[1] / 100.0 * hh;
            drawLine(1, y, 1, hh);
            copyArea(0, 0, ww, hh, 1, 0);
            return;
        break;
        case "config":
            /* initialization code, prepare the variables... */
            usual();
            clearWindow();
            ww = get("width");
            hh = get("height");
            return;
        break;
    }
    usual();
}

\gapV {2}
\gapH {2}
\width {100}
\height {50}
\BGCOLOR {blue}
\FGColor {white}
\border {6}
\
\name {monitorBar.socket}
\parent {monitorBar}
\class {socket}
\host {pebble.berkeley.ora.com}
\port {7777}
\script {
    /* sym share price
     */
    switch (arg[0]) {
        case "input":
            /* there's data in the queue, read and parse the data,
             * then relay the information to the drawing object
             */
            data = input();
            send(parent(), "graph", int(nthWord(data, 3)));
            return;
        break;
        case "init":
            /* tell viola to notify this object when data stream
             * has enough useful information (when there's a
             * carriage return character) for this object to
             * read the input queue.
             */
            usual();
}

```

```
    set("outDelimStr", "\r\n");
    set("inDelimStr1", '\n');
    startClient();
    return;
}
break;
}
usual();
}
\
```

---

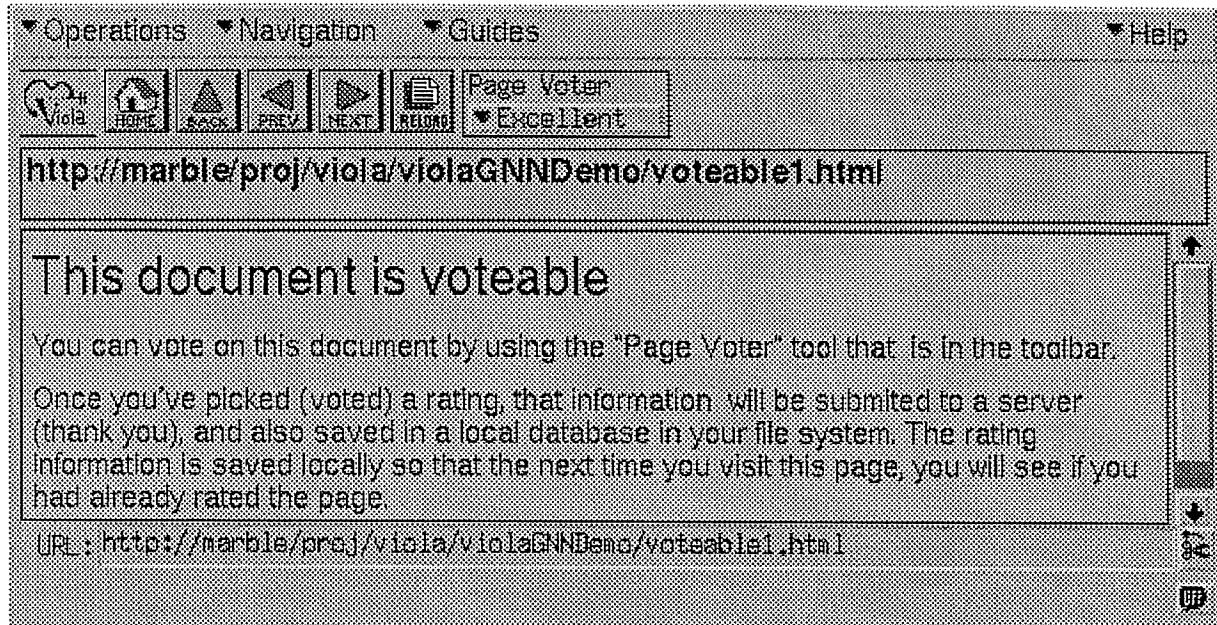
This document can be found at:

[http://marble.ebay.gnn.com/proj/viola/violaGNNDemo/monitorBarDemo\\_out.html](http://marble.ebay.gnn.com/proj/viola/violaGNNDemo/monitorBarDemo_out.html)

*pei@gnn.com*

# Embedding Viola Applets in the ToolBar

## Example: Page Voter



This document is voteable. You can vote on this document by using the "Page Voter" tool that is in the toolbar.

Once you've picked (voted) a rating, that information will be submitted to a server (thank you), and also saved in a local database in your file system. The rating information is saved locally so that the next time you visit this page, you will see if you had already rated the page.

FYI, The "Page Voter" applet is linked to this page using the following markup. Note that same applet can be reused by different pages (only the 'ARG' attribute is changed to identify the page being voted on).

```
<LINK REL="tool" HREF="voter2.v" ARG="voteable1.html">
```

And here's the "voter2.v" applet listing:

```
\class {vpane}
\name {voter2}
\children {voter2.label voter2.rating}
\script {
    switch (arg[0]) {
        case "make":
            votingOn = arg[2];
            info = send("ratedUrlDB", "fetch", votingOn);
            if (info) send("voter2.rating", "setRate", int(info[0]));
    }
}
```

```

        return;
break;
case "submit":
    print(">>>>>> SUBMIT: doc = {", votingOn, "} rating = ",
        arg[1], "\n");

    HTTPGet(concat("http://marble.ebay.gnn.com/cgi-bin/votepg?",
                    votingOn, "=", arg[1]));
    send("ratedUrlDB", "put", votingOn, arg[1], "description...");
    send("ratedUrlDB", "flush");
    return;
}
break;
}
usual();
}

\maxWidth {100}
\border {6}
\gapH {2}
\gapV {2}
\
\class {txtLabel}
\name {voter2.label}
\parent {voter2}
\label {Page Voter}
\paneConfig {westToEast}
\script {
    switch (arg[0]) {
        case "hint":
            return "Vote on the document!";
        break;
        case "enter":
        case "leave":
            send("www.msg.tf", "suggest", self(), arg[0]);
        break;
    }
    usual();
}
\gapH {2}
\
\class {menu}
\parent {voter2}
\name {voter2.rating}
\menuConfig {
. {Excellent}   {send(self(), "rating", 4, "Excellent");}
. {Good}         {send(self(), "rating", 3, "Good");}
. {Average}     {send(self(), "rating", 2, "Average");}
. {Poor}         {send(self(), "rating", 1, "Poor");}
}
\label {unrated}
\script {
    switch (arg[0]) {
        case "rating":
            set("label", arg[2]);
            render();
    }
}
```

```
    send(parent(), "submit", arg[1]);
    return;
break;
case "setRate":
    switch (arg[1]) {
        case 1: set("label", "Poor"); break;
        case 2: set("label", "Average"); break;
        case 3: set("label", "Good"); break;
        case 4: set("label", "Excellent"); break;
    }
    render();
    return;
break;
case "hint":
    return "Choose a rating...";
break;
}
usual();
}
\border {3}
\gapH {2}
\
```

---

This document can be found at:

<http://marble.ebay.gnn.com/proj/viola/violaGNNDemo/aboutVoteablePages.html>

*pei@gnn.com*